Robot Programming Practice #2

Dept. of Mech. Robotics and Energy Eng.
Dongguk University

External Power Supply

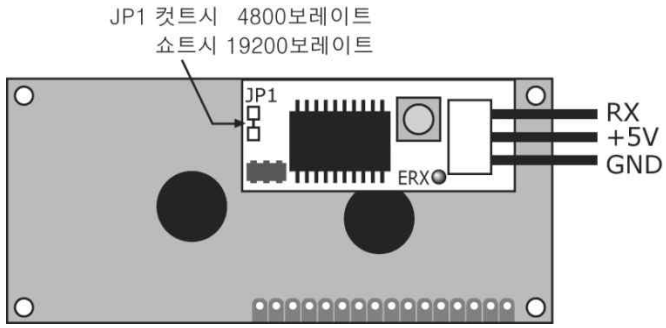
- We need an external power supply to drive high-power devices such as LCD and motor driver.
- We are going to use a DC 5V adaptor.

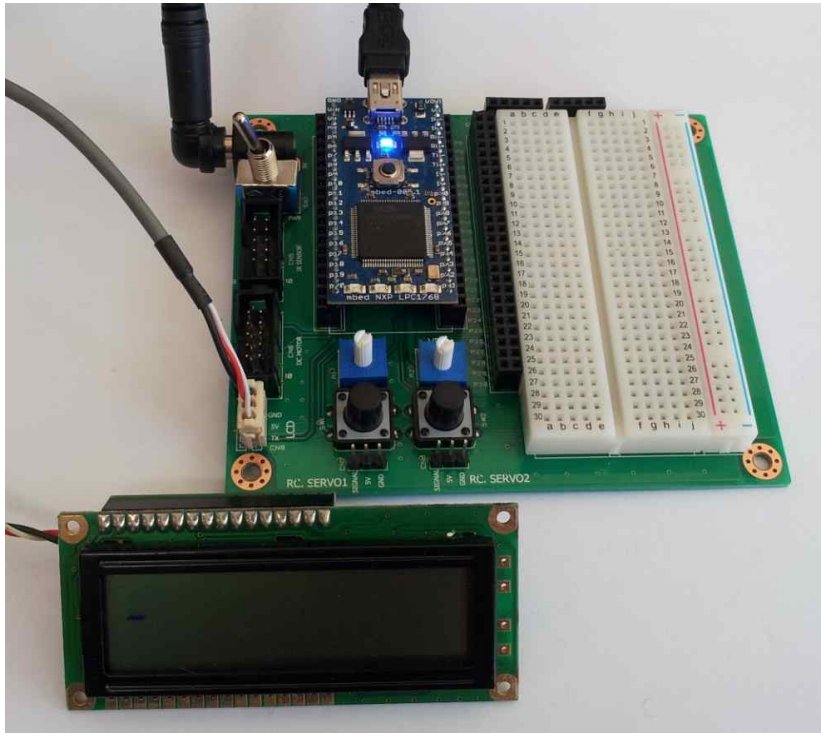
Connecting a LCD display

- Let's connect the adaptor to the mbed SB and LCD.
- We will use a serial English LCD module.

Front

Back





Serial LCD display

- The LCD module needs a 5V power. On-board power of the mbed is not enough to drive the LCD module.
- A RX pin of the LCD module is connected to pin 28.
- The LCD module used in our course has the following protocol.

Command(Hex)	Usage
A0	Initialize LCD. At least 10ms delay time is necessary after sending out this command.
A1 X Y	Assign the display position.
A2 String 0	Display string on the LCD. 0 should be followed.

Sending out string to the LCD

- Write the following program, compile it and download.

```
#include "mbed.h"
//--- Serial LCD (Comfile) part ELCD162-BL
//-------- LCD 2x16 ----------
// MBED p28-------------------RX
// MBED Vout------------------5V
// MBED GND-------------------GND
// --------------------------

Serial lcd(p28, p27);       // tx, rx

int main() {
    lcd.baud(19200);
    lcd.putc(0xA0); //  Initialize LCD
    wait(0.2);
//  Print character
//  Position (column,row)
    lcd.printf("%c%c%c",0xA1,2,0);
    lcd.printf("%cHello World!%c",0xA2,0x00);
}
```

A Closer Look

```
Serial lcd(p28, p27);        // tx, rx
```

- Serial is a generic protocol used by computers and electronic modules to send and receive control information and data.
- tx represents the transmission and rx represents the reception.
- lcd is defined as the serial communication variable with tx and rx.

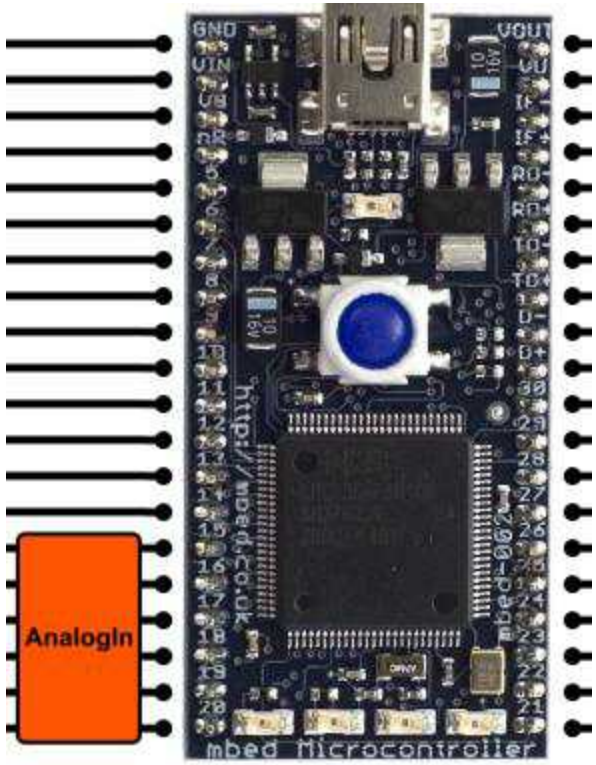
A Closer Look

```
lcd.baud(19200);
lcd.putc(0xA0); //  Initialize LCD
lcd.printf("%c%c%c",0xA1,2,0);  //  Position (column,row)
lcd.printf("%cHello World!%c",0xA2,0x00);
```

- Baud rate is set 19200.
- putc command sends out data to tx.
- printf command can send out many characters at the same time.

Analog inputs on the mbed

- The mbed has up to six analog inputs, on pins 15 to 20.



Reading analog inputs(Pot)

- Use the on-board potentiometer No.1(which is already connected to pin 19).
  - Start a new mbed project and enter the code below.
  - This code will continuously display the analog input value when used with the LCD.

Reading analog Inputs(Pot)

```
#include "mbed.h"
//--- Serial LCD (Comfile) part ELCD162-BL
//-------- LCD 2x16 ----------
// MBED p28------------------RX
// MBED Vout-----------------5V
// MBED GND------------------GND
// --------------------------
Serial lcd(p28, p27);       // tx, rx
AnalogIn Ain(p19);
float ADCdata;
int main() {
    lcd.baud(19200);
//  Initialize
    lcd.putc(0xA0);
    wait(0.2);
//  Print character
    while(1){
    lcd.printf("%c%c%c",0xA1,0,0);
    ADCdata = Ain;
    lcd.printf("%cADC Data: %f %c",0xA2,ADCdata,0x00);
    wait(0.5);
   }
}
```

Concepts of DA conversion

- We can represent the digital-to-analog convertor (DAC) as a block diagram with a digital input, D, and an analog output, vo.
- The output range of the DAC is the difference between the maximum and minimum output voltages, i.e.

$$V_r = V_{\max} - V_{min}$$

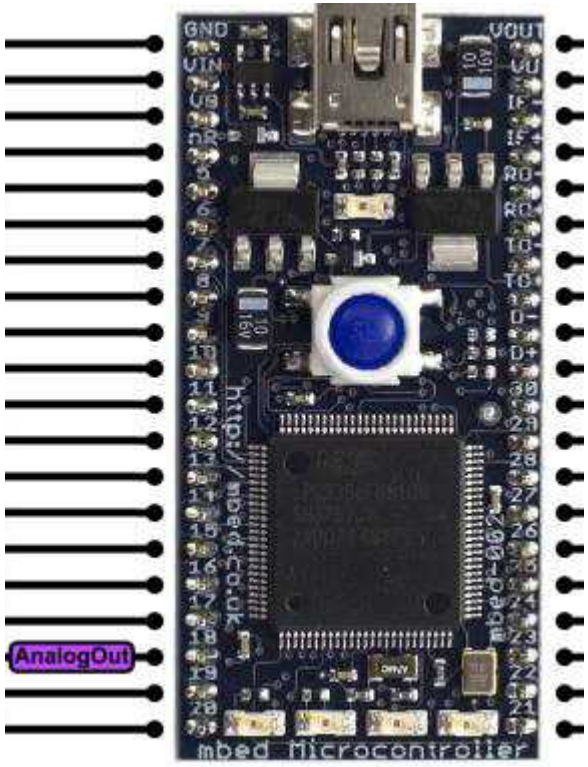
- The particular output range is usually defined by a fixed voltage reference supplied to the DAC
- Digital control lines allow a microcontroller to setup and communicate with the DAC

Concepts of DA conversion

- The mbed's LPC1768 chip has a 10-bit DAC (i.e. n=10).
- The mbed uses its own 3.3 V power supply as voltage reference.
- There will therefore be 2n steps in the mbed DAC output characteristic, i.e. 1024.
- The step size, or resolution, is therefore be 3.3/1024, i.e. 3.2mV per bit.

Concepts of DA conversion

- The mbed has a single analog output on pin 18.



Analog Output with the mbed

- The mbed analog output on pin 18 is configured by the following declaration:

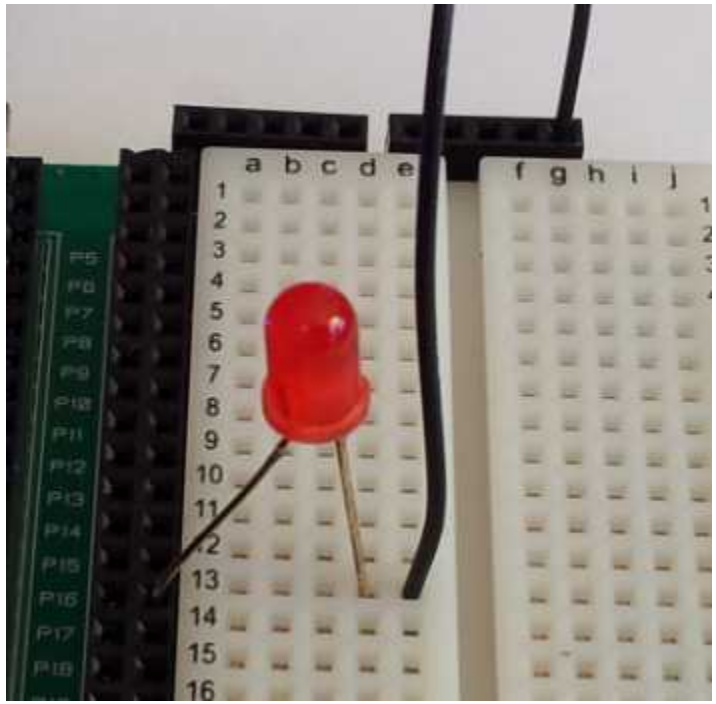
```
AnalogOut Aout(p18);
```

- By default, the analog object takes a floating point number between 0.0 and 1.0 and outputs this to pin 18
- The actual output voltage on pin 18 is between 0V and 3.3V, so the floating point number that is output as a voltage is scaled by a factor of 3.3

Analog Output with the mbed

- Attach a LED to p18, compile the program shown below and familiarize yourself with the analog output.

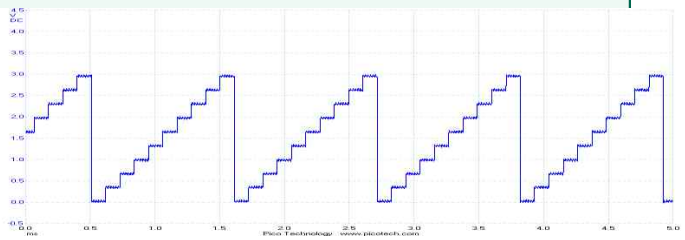
```
#include "mbed.h"

AnalogOut Aout(p18);

int main() {
    while(1) {
        Aout = 0.25;
        wait(0.5);
        Aout = 0.5;
        wait(0.5);
        Aout = 0.75;
        wait(0.5);
        Aout = 1.0;
        wait(0.5);
    }
}
```



Analog Output with the mbed

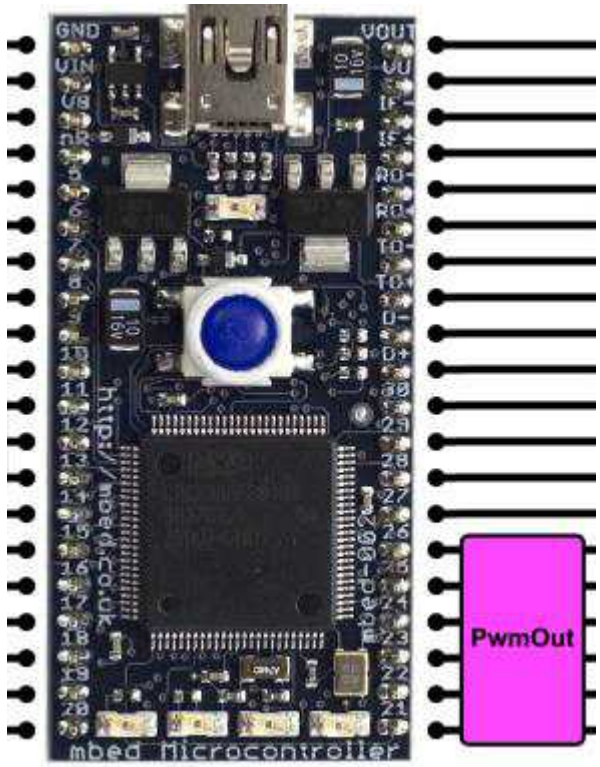
- Now make a sawtooth wave and view it on the LED. Create a new program and enter the following code.

```
#include "mbed.h"

AnalogOut Aout(p18);
float y;

int main() {
    while(1) {
        for (y=0.0; y<1.0;y=y+0.1)
        {
        Aout = y;
        wait(0.1);
        }
    }
}
```



PWM on the mbed

- The PwmOut interface is used to control the frequency and mark-space ratio of a digital pulse train.
- The mbed has up to six PWM outputs, on pins 21 to 26, although these PwmOuts all share the same period timer.



PWM on the mbed

- This example code uses a pulse width modulation signal to increase and decrease the brightness of the onboard LED

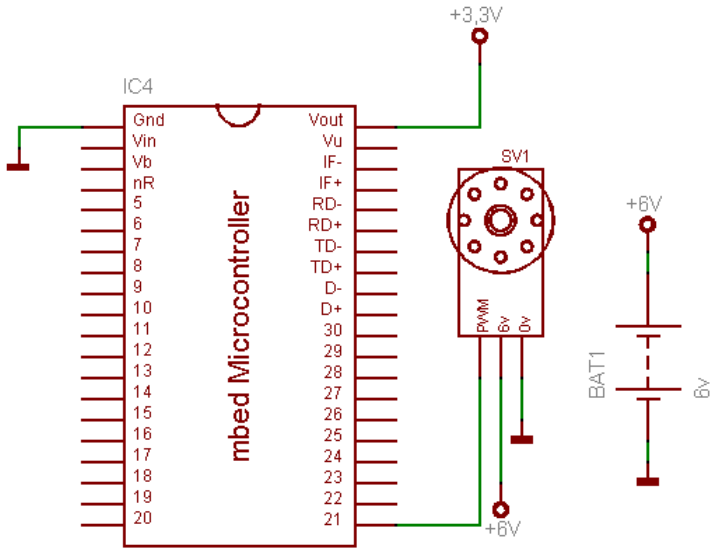
```
#include "mbed.h"

PwmOut led(LED1);

int main() {
    while(1) {
        for(float p = 0.0f; p < 1.0f; p += 0.1f) {
            led = p;
            wait(0.1);
        }
         for(float p = 0.9f; p > 0.1f; p -= 0.1f) {
            led = p;
            wait(0.1);
        }
    }
}
```

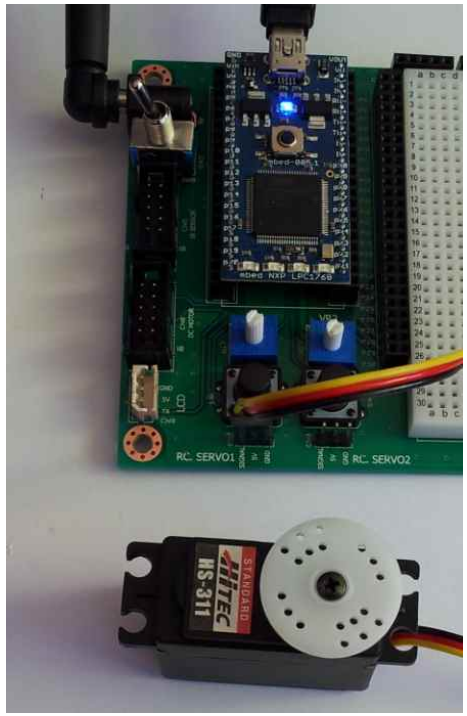
RC Servo Motor

- A RC servo is a small rotary position control device, used for example in radio-controlled cars and aero planes to position controllers such as steering, elevators and rudders.
- It was made to control the position of the steering wheel in a RC car and a RC airplane.
- Recently, it is popularly used in robots.

Controlling servo position

- Connect the servo to RC SERVO1(p21) of the SB.
- The servo requires a higher current than the USB standard can provide, and so it is essential that you power the servo using an external power supply.





Controlling the servo position

- This example code uses a Servo library to sweep a servo through its full range. Check the home page (Servomotor under Components).

```
#include "mbed.h"
#include "Servo.h"

Servo myservo(p21);

int main() {
    for(float p=0; p<1.0; p += 0.02) {
        myservo = p;
        wait(0.1);
    }
    myservo = 0;
    wait(0.2);
     for(float p=0; p<1.0; p += 0.1) {
         myservo = p;
         wait(0.2);
    }

}
```